

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Model hexapodu .....	5
2.2 Základní principy chůze.....	6
2.3 Chůze v terénu.....	8
2.4 Inverzní kinematika.....	9
2.5 Křivky pro plánování trajektorií.....	9
2.6 Existující projekty .....	11
<b>3 Konkrétní hexapod</b>	<b>13</b>
3.1 Konstrukce.....	13
3.2 Elektronika .....	15
<b>4 Návrh</b>	<b>21</b>
4.1 Návrh rozhraní knihovny.....	21
4.2 Řízení chůze.....	22
4.3 Odvození inverzní kinematiky .....	29
<b>5 Implementace</b>	<b>33</b>
5.1 Programovací jazyk .....	33
5.2 Složení aplikace .....	33
5.3 Interface knihovny .....	34
5.4 Konfigurace robota .....	35
5.5 Průběh funkce pro chůzi.....	36
5.6 Chůze v terénu.....	36
5.7 Implementace inverzní kinematiky.....	37
5.8 Tvorba trajektorie těla a nohou.....	37
5.9 Synchronizace nohou.....	38
5.10 Komunikace se servo kontrolerem.....	38
5.11 Zprovoznění akcelerometru.....	39
5.12 Použití klepet .....	39
5.13 Demo aplikace.....	40
<b>6 Testování</b>	<b>41</b>
6.1 Chůze po rovině.....	41
6.2 Chůze v terénu .....	42

6.3 Přenášení předmětů.....	43
<b>Závěr</b>	<b>45</b>
<b>Literatura</b>	<b>47</b>
<b>A Uživatelská příručka</b>	<b>51</b>
A.1 Zapnutí robota .....	51
A.2 Dostupné funkce knihovny.....	52
A.3 Ukázka použití knihovny .....	54
<b>B Zapojení akcelerometru</b>	<b>55</b>
<b>C Seznam použitých zkratk</b>	<b>57</b>
<b>D Obsah přiloženého CD</b>	<b>59</b>

---

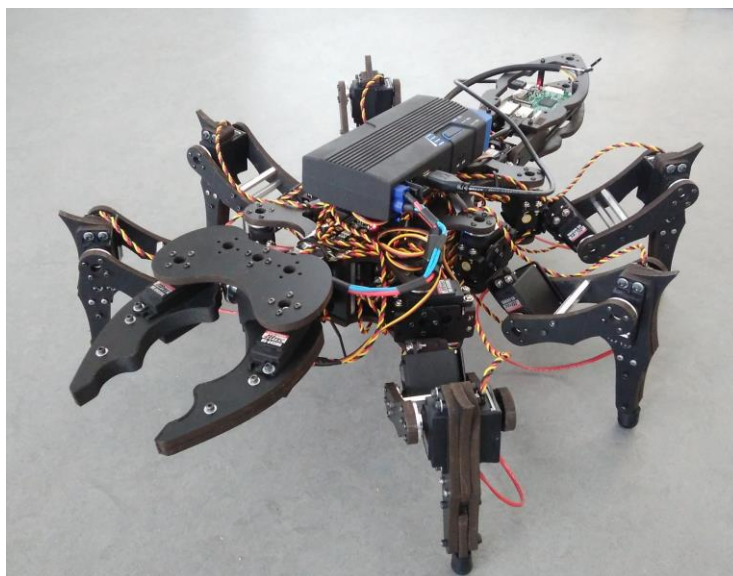
## Seznam obrázků

0.1	Šestinohý robot A-Pod od firmy Lynxmotion .....	1
2.1	Tvary hexapodů .....	6
2.2	Typy chůze .....	7
2.3	Typy křivek a jejich řídicí body.....	10
3.1	A-Pod od firmy Lynxmotion .....	13
3.2	Rozměry těla A-Podu .....	14
3.3	Rozměry nohy A-Podu .....	15
3.4	Schéma zapojení elektroniky.....	15
3.5	Umístění elektroniky .....	16
3.6	Indexy pinů, do kterých jsou zapojena serva a dotykové senzory .....	17
3.7	Struktura příkazu pro pohyb serva .....	18
3.8	Průběh PWM signálu .....	19
3.9	Dotykový senzor a jeho umístění.....	20
4.1	Schéma cyklů nohy.....	22
4.2	Schéma chůze po přímce.....	24
4.3	Schéma pro výpočet vzdálenosti a úhlu konce nohy od středu těla	25
4.4	Schéma krajní polohy nohy při otáčení na místě .....	26
4.5	Návrh Bézierovy křivky .....	27
4.6	Chůze po křivce: Vyjádření souřadnic půlkroku relativně k poloze těla po posledním půlkroku .....	28
4.7	Inverzní kinematika - půdorys.....	29
4.8	Inverzní kinematika - bokorys .....	30
6.1	Chůze robota po nakloněné rovině .....	42
6.2	Přenášení plechovky pomocí klepet.....	43
A.1	Popis zapojení baterie.....	51

B.1	Raspberry Pi - čísla pinů .....	56
B.2	Zapojení I <sup>2</sup> C na RPi .....	56

## Úvod

Robotika je v dnešní době velmi rozvíjené odvětví, které se postupně dostává do našich životů. Ať už je to v podobě domácích robotů pro úklid a vaření nebo velkých automatizovaných linek, které zastávají těžké monotonní práce. Své využití našli roboti i při plnění nebezpečných úkolů. Příkladem může být zneškodňování trhavin nebo otvírání podezřelých zavazadel.



Obrázek o.1: Šestinohý robot A-Pod od firmy Lynxmotion

Pro práci na nepřístupných místech, jako jsou sutiny nebo cizí planety, se nejvíc hodí kráčiví roboti, kteří dokáží překonat větší překážky a nerovnosti terénu než koloví nebo pásoví roboti. Zejména vhodný je šestinohý robot – zvaný také hexapod – protože má ideální počet nohou, který při vhodné synchronizaci končetin zajišťuje dobrou stabilitu při rychlém i pomalém pohybu.

Cílem práce je vytvořit programové vybavení pro šestinohého robota A-Pod od firmy Lynxmotion, které umožní jeho pohyb. Řešení bude zpracováno ve formě knihovny.

Cílem rešeršní částí je seznámení s modelem hexapodu a principem jeho chůze. Dále bude popsána kostra a elektronika konkrétního hexapodu, pro který je řešení vyvíjeno.

Cílem praktické části je navrhnout a implementovat knihovnu, která bude obsahovat funkce pro pohyb robota. Robot se bude umět pohybovat po rovině ve všech směrech a otáčet se na místě. Dále za pomoci dotykových senzorů na nohách a akcelerometru bude chodit v terénu. Řešení bude obsahovat část, která umožní pohyb klepet na hlavě a manipulaci s předměty jejich pomocí. Na robotovi bude zprovozněna kamera. Všechny funkce řešení budou otestovány a bude k nim vytvořena demo aplikace.

## KAPITOLA 2

# Analýza

### 2.1 Model hexapodu

Hexapod je šestinohý robot. Spolu s kvadrupenty a humanoidními roboty se řadí mezi kráčivé roboty. Oproti robotům na kolech mají menší potenciální rychlost na rovině, ale vynikají při pohybu v terénu, který je pro jezdící roboty špatně přístupný [1]. Další výhodou je, že méně narušují povrch, po kterém se pohybují.

#### 2.1.1 Fyzické parametry hexapodů

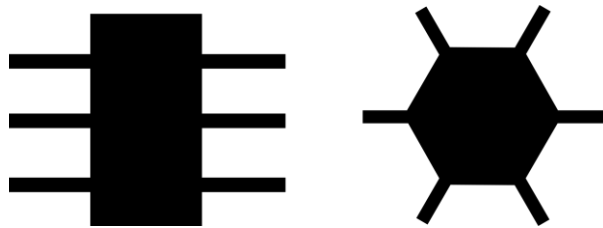
Podle tvaru se hexapody dělí na dvě základní skupiny:

1. Obdélníkové hexapody mají tělo tvaru obdélníku a na každé z delších protistran mají tři nohy. Je u nich snazší koordinace nohou při pohybu rovně, protože pohyby pro každou nohu mohou být identické. Avšak pro otáčení robota musí být implementován samostatný algoritmus. [2]
2. Hexagonální roboti mají kruhové nebo hexagonální tělo, ze kterého symetricky od středu těla paprskovitě vychází nohy. V základním postavení mají hexagonální roboti mnoho výhod. Dokážou se pohybovat do všech směrů stejným způsobem bez nutnosti otáčet se. Další velkou výhodou je větší stabilita a její snazší udržení při různých pohybech. [3]

Jiným parametrem, kterým se hexapody od sebe liší, je počet stupňů volnosti. „Počet stupňů volnosti robota je nejmenší počet reálných parametrů

*potřebných k reprezentaci jeho konfigurace.*“ [4]

Každá z končetin robota je rozdělena na části, které jsou mezi sebou spojeny klouby. Existuje mnoho druhů kloubů. Lze je dělit podle druhu pohybu nebo podle počtu stupňů volnosti. Pro náš případ jsou důležitá serva, která mají rotační pohyb s jedním stupněm volnosti. Při popisu hexapodů se buď



Obrázek 2.1: Tvary hexapodů: obdélníkový (vlevo) a hexagonální

uvádí celkový součet počtu stupňů volnosti nebo je vyjádřen ve formě počtu stupňů volnosti na jednu nohu.

Aby byl umožněn pohyb nohy, musí mít alespoň jeden stupeň volnosti. V tomto případě se většinou jedná o spoj nohy a těla, který umožňuje nepřetržitě otáčení o  $360^\circ$  [5]. Tito roboti jsou často označováni RHex.

Zároveň počet stupňů volnosti větší než šest je nadbytečný, protože s šesti stupni je možné konec nohy dát do jakékoli polohy (včetně koncového úhlu), která je v dosahu [6]. Nejčastěji se setkáme s hexapody, které mají dva nebo tři stupně volnosti na jednu nohu.

## 2.2 Základní principy chůze

### 2.2.1 Cyklus jedné nohy

Každá noha se většinou pohybuje v cyklu. Ten je možné rozdělit na dvě části - podpůrnou a přesouvací. V podpůrné fázi je noha na zemi, zajišťuje stabilitu robota a při pohybu se posouvá směrem proti pohybu. Robot se od ní odráží a tím se posouvá dopředu. Aby bylo možné provádět pohyb v podpůrné fázi, je třeba zařadit fázi přesouvací, kdy se noha zvedne do vzduchu a přesune se ve směru pohybu. Zde se položí na zem a přejde zpět do podpůrné fáze. [2]

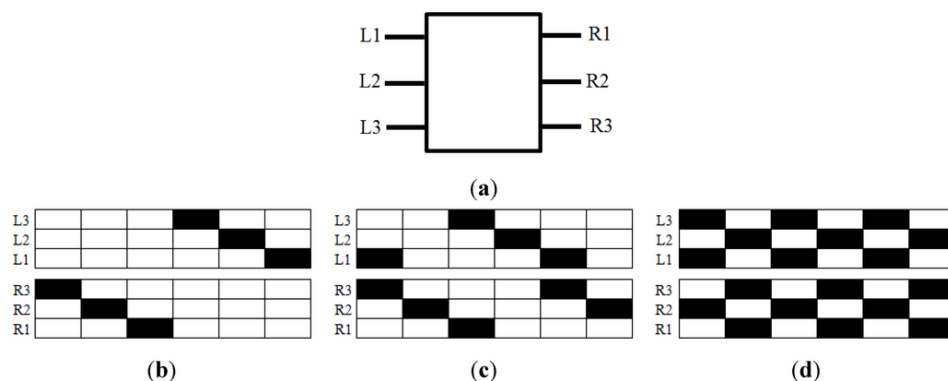
Poměr času podpůrné fáze ku celkovému času kroku nazýváme faktorem střídání (duty factor). Jeho hodnoty se pohybují od nuly do jedné. Faktor střídání omezuje maximální možnou rychlost, které je robot schopný dosáhnout. [2, 7]

### 2.2.2 Stabilita

Robot je staticky stabilní, jestliže se jeho těžiště nachází v mnohoúhelníku tvořeném jeho nohama, které jsou v kontaktu se zemí. Aby bylo možné dosáhnout statické stability, tak musí být na zemi alespoň tři nohy. Také platí, že čím blíže je těžiště ke středu mnohoúhelníku, tím je robot stabilnější. Hexapod využívá statické stability, protože k ní má díky šesti nohám ideální předpoklady. Hexapod může mít při chůzi snížený faktor střídání až na 0,5 a přitom má stále na zemi tři nohy, které stačí k dosažení statické stability. [2]

### 2.2.3 Typy chůze (synchronizace nohou)

Hexapod má díky šesti nohám velkou volnost při volbě synchronizace nohou. Jediným omezením je umístění alespoň tří nohou na zemi, aby neporušil podmínky statické stability. Následuje výčet a popis nejpoužívanějších typů chůze. Indexace nohou použitá při popisu synchronizací je znázorněna na obrázku 2.2. [5, 3]



Obrázek 2.2: Typy chůze a) indexace nohou pro účely popisu synchronizace nohou b) typ vlna c) typ vlnění d) typ tripod [5]

#### Tripod (3+3)

Při využití tohoto systému jsou nohy rozděleny do dvou skupin - L1L3R2 a L2R1R3. Nohy ve stejné skupině mají stejnou fázi cyklu chůze. Většinu doby je jedna skupina v podpůrné fázi a druhá v přesouvací. Když se přechází z fáze do fáze, existuje krátký okamžik, kdy se všechny nohy dotýkají země. Za jeden cyklus jsou udělány dva kroky. Faktor střídání je  $\frac{1}{2}$ .

#### Vlna (5+1)

Každá noha má jinou fázi. Vždy je ve vzduchu nejvýše jedna. Tím je docíleno maximální stability při pohybu. Tento způsob je vhodný do členitého terénu a na překonávání překážek. Je mnoho možností, jak mohou nohy následovat za sebou. Faktor střídání je  $\frac{5}{6}$ .

#### Alternující tripod

Jedná se o spojení vlny a tripodu. Rozdělení nohou je stejné jako v tripodu, avšak nohy nejsou zvedány synchronně, ale jsou fázově posunuty. Změnou délky překrytí přesouvacích fází je možné docílit větší stability na úkor maximální rychlosti nebo i opaku. Faktor střídání se pohybuje od  $\frac{1}{2}$  do  $\frac{5}{6}$ .



### 4+2

Nohy jsou rozděleny do párů  $L_1R_1$ ,  $L_2R_2$ ,  $L_3R_3$ . Ve vzduchu je vždy jeden z párů. Začíná se od zadní dvojice. Faktor střídání je  $\frac{2}{3}$ .

### Vlnění

Napodobuje chůzi čtyřnohých robotů, avšak díky většímu množství nohou je stále stabilní. Nohy jsou rozděleny do čtyř skupin -  $L_1R_3$ ,  $R_2$ ,  $L_3R_1$ ,  $L_2$ . V přesouvací fázi je vždy jen jedna ze skupin a střídají se v uvedeném pořadí. V každou chvíli jsou tedy na zemi minimálně 4 nohy. Faktor střídání je  $\frac{3}{4}$ .

#### 2.2.3.1 Porovnání typů chůze

Nejvyšší potenciální rychlost má tripod. Navíc je lehce implementovatelný, a proto je na rovné povrchy bez překážek jedním z nejlepších řešení. Oproti ostatním synchronizacím má však horší stabilitu, což ale vadí jen ve členitém terénu.

Další variantou je vlnění, které je o něco stabilnější, ale platí za to nižší rychlostí.

Lepším řešením může být alternující tripod, který se velice dobře dokáže interaktivně přizpůsobovat, protože kromě úpravy rychlosti kroku můžeme upravovat i délku překrytí přesouvacích fází. Proto se hodí do středně náročného terénu.

Pro velice nepravidelný terén nebo terén s velkými překážkami je dobrým řešením vlna, která má nejlepší stabilitu a její nízká rychlost v tomto prostředí není vadou.

## 2.3 Chůze v terénu

Chůze v terénu je mnohem náročnější než pohyb po rovině, proto se robot musí pohybovat pomaleji. Je zapotřebí, aby upravoval pohyby jednotlivých nohou podle tvaru terénu, po kterém jde. Abychom mohli úpravu provést, potřebujeme informace o okolí. Ty můžeme získat například jako předem připravenou 3D mapu [1], ale častějším způsobem je použití senzorů v průběhu chůze robota.

#### 2.3.1 Změny vůči pohybu po rovině

Při pohybu po rovině stačí, aby se tělo posouvalo dopředu. To znamená, že nohy, které se dotýkají země, jdou dozadu. Tento způsob však není dostačující pro pohyb v terénu, protože tělo v tom případě zůstává ve stejné absolutní výšce. Což znamená, že pokud bychom robota nechali tímto způsobem jít nahoru po nakloněné rovině, tak po pár krocích narazí tělem do podložky. Tomu je potřeba předejít a pravidelně upravovat vzdálenost těla a podložky.

### 2.3.2 Poloha těla vůči zemi

Tělo může být nastaveno paralelně se zemí, nebo může být udržováno vodorovně, například pomocí akcelerometru. Způsob s paralelním postavením těla lze použít pouze na nakloněné rovině s malým úhlem a vyznačuje se tím, že se nohy přesouvají stejně, jako by byl hexapod na rovné zemi. Když je tělo vodorovně, zvládne udržet rovnováhu i na strmějších nepravidelných terénech. [3]

## 2.4 Inverzní kinematika

Inverzní kinematika je v robotice hojně využívaný matematický postup. Jedná se o opačný proces od dopředné kinematiky, který podle natočení serv jednoduše vypočítá polohu koncového bodu. Tedy pomocí inverzní kinematiky najdeme natočení serv takové, aby byl koncový bod nohy na požadovaných souřadnicích. Inverzní kinematika v řadě případů není jednoznačná, tedy pro jedny souřadnice koncového bodu existuje více kombinací natočení serv, které je splňují. [4]

Základním přístupem inverzní kinematiky je geometrické vyjádření. Tento postup je aplikovatelný na systémy, které nemají velký počet stupňů volnosti. Při vyšším počtu začíná být příliš složitý.

Geometrický výpočet využívá především goniometrické funkce, Pythagorovu a Kosinovou větu. Odvození vzorců pro konkrétní hexapod je uvedeno v sekci 4.3.

## 2.5 Křivky pro plánování trajektorií

Pro plánování trajektorií robotů a jejich končetin se používají křivky, protože díky nim lze dosáhnout plynulejšího a tím i přirozenějšího pohybu. V této práci jsou využity kubické Bézierovy křivky pro plánování trajektorie těla jako celku.

Křivku lze fyzikálně chápat jako dráhu pohybujícího se bodu, jehož souřadnice se mění v čase. V tomto případě je křivka vyjádřena parametricky v závislosti na parametru  $t$  (čase), který se většinou pohybuje v intervalu  $\langle 0, 1 \rangle$ .

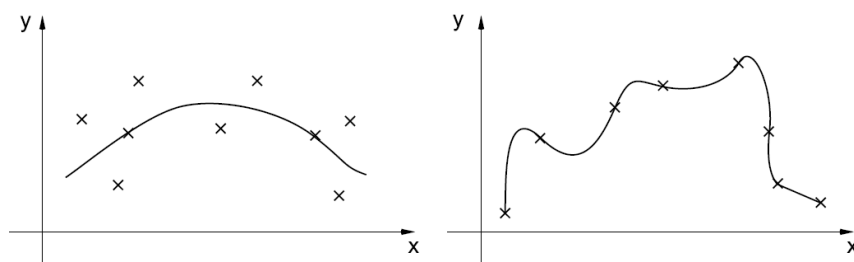
Rovnice 2.1 zobrazuje vyjádření tečného vektoru v bodě  $Q(t_0)$ . Vektor je určen derivací parametricky vyjádřené křivky po složkách.

$$\vec{q}'(t_0) = (x'(t_0), y'(t_0), z'(t_0)) = \left( \frac{dx(t_0)}{dt}, \frac{dy(t_0)}{dt}, \frac{dz(t_0)}{dt} \right). \quad (2.1)$$

Při modelování křivek se využívají takzvané řídicí body, kterými je ovlivňován průběh křivky. Existují dva základní typy křivek vzhledem k interpretaci řídicích bodů:

## 2. ANALÝZA

- aproximační křivky jsou tvarovány na základě řídicích bodů, ale nemusí jimi procházet.
- interpolační křivky přímo probíhají řídicími body,



Obrázek 2.3: Aproximační (vlevo) a interpolační křivka a jejich řídicí body [8]

Křivky můžeme na sebe postupně napojovat. Bod, ve kterém se dvě křivky spojují, nazýváme uzlem. Důležitým faktorem při spojování křivek je to, jak dobře na sebe navazují. Tuto informaci můžeme vyjádřit spojitostí.

Rovnice 2.1 až 2.4 byly převzaty z knihy [8]. Ze stejného zdroje byly čerpány i informace pro tuto kapitolu.

### 2.5.1 Bézierova křivka

Bézierova křivka se řadí mezi aproximační křivky, ale její zvláštností je, že prochází počátečním a koncovým řídicím bodem.

Rovnice 2.2 zobrazuje vyjádření Bézierovy křivky  $n$ -tého stupně. K určení křivky  $n$ -tého stupně je zapotřebí  $n+1$  řídicích bodů ( $P_i$ ). Pro výpočet křivky  $n$ -tého stupně je použit Bernsteinův polynom  $n$ -tého stupně ( $B_i^n$ ), jehož vyjádření je v rovnici 2.3.

$$Q(t) = \sum_{i=0}^n B_i^n(t) P_i \quad (2.2)$$

$$B_i^n = \binom{n}{i} t^i (1-t)^{n-i}; t \in \langle 0, 1 \rangle, i = 0, 1, \dots, n. \quad (2.3)$$

V rovnici 2.4 je uveden maticový zápis výpočtu Bézierovy kubiky, který dále v práci využívám.

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (2.4)$$

## 2.6 Existující projekty

Existuje mnoho projektů, které se zaměřují na pohyb šestinohého robota. Každý z nich je ale uzpůsoben pro konkrétní fyzický model hexapodu, kód je implementován na danou elektroniku a většinou ho není možné použít na jiného robota. Dalším aspektem, ve kterém se implementace liší, je volba ovládání robota – zda je robot řízen pomocí ovladače či mobilu, nebo je alespoň do jisté míry autonomní. Dále jsou návrhy přizpůsobeny sensorům, které jsou na hexapodu přítomny.

Při provádění analýzy existujících řešení jsem nenalezla žádný projekt zaměřený na A-Pod s elektronikou, kterou mám k dispozici. Při odhlédnutí od elektroniky byl nejbližší projekt [9] s ovládáním ovladačem (bez zveřejněného kódu).

Článek [1] představuje robota, který chodí terénem s pomocí stereo kamery. Na základě dat z kamery hexapod upravuje výšku kroku, délku kroku a tvrdost došlapu, aby byl průchod terénem co nejhladší. Pokud se robot vyskytne v prostředí jako je vysoká tráva, kde není možné využít kameru, tak jako zpětnou vazbu z prostředí používá senzory v nohách.

Chůze hexagonálních hexapodů je hlavním tématem článku [3]. Autoři probírají různé typy chůzí, jejich parametry a použitelnost. Kromě základních typů chůzí představují i ty odolné vůči poruchám.

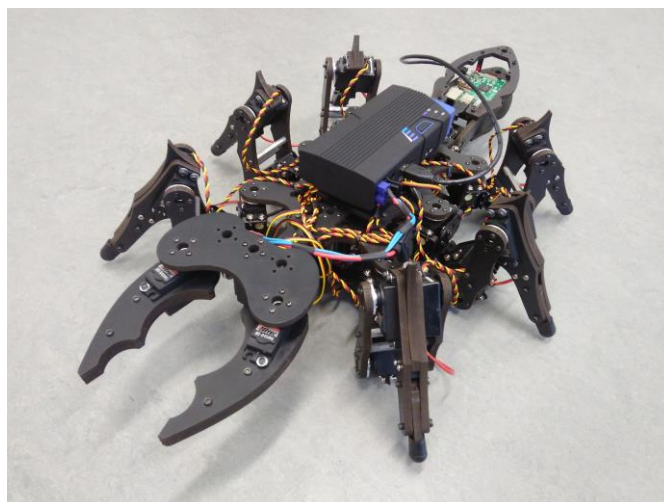
V projektu [10] hexapod chodí a tančí, přičemž je řízen pomocí mobilní aplikace. Hlavním parametrem funkcí pro pohyb robota je v této práci časový údaj, který specifikuje, po jak dlouhou dobu má hexapod pohyb vykonávat.

Phoenix code je označení projektu vytvářeného komunitou. [11] Projekt je velmi rozsáhlý. Původně byl napsán v programovacím jazyce Basic, ale v mezidobí byl již převeden do C/C++ na platformu Arduino. Kód v C/C++ je specificky vytvořený pro hexapody PhantomX – model MK-I a MK-II a řízení robota probíhá přes ovladač ArbotiX.

## Konkrétní hexapod

V této kapitole popíši konkrétní hexapod, pro který knihovnu vyvíjím. Jedná se o model A-Pod od firmy Lynxmotion.

V první části se zaměřím na jeho kostru a mechanické vlastnosti. Ve druhé části popíši elektroniku, její parametry, možnosti a zapojení.



Obrázek 3.1: A-Pod od firmy Lynxmotion

## **3.1 Konstrukce**

### **3.1.1 Tvar**

Tento model nespadá ani do jedné ze základních skupin hexapodů vzhledem k tvaru těla, ale má blíže k obdélníkovým. Jeho tělo je tvaru obdélníku, jeho nohy jsou umístěné po stranách, ale přední a zadní pár musí být vytočen od

### 3. KONKRÉTNÍ HEXAPOD

---

středního páru, aby bylo s nohama možné pohybovat. Toto je zohledněno už při montáži robota. Středová poloha serv zajišťujících horizontální pohyb na krajních párech nohou (serva nejbliže u těla) by měla být vytočena o  $45^\circ$  od středového páru.

#### 3.1.2 Stupně volnosti

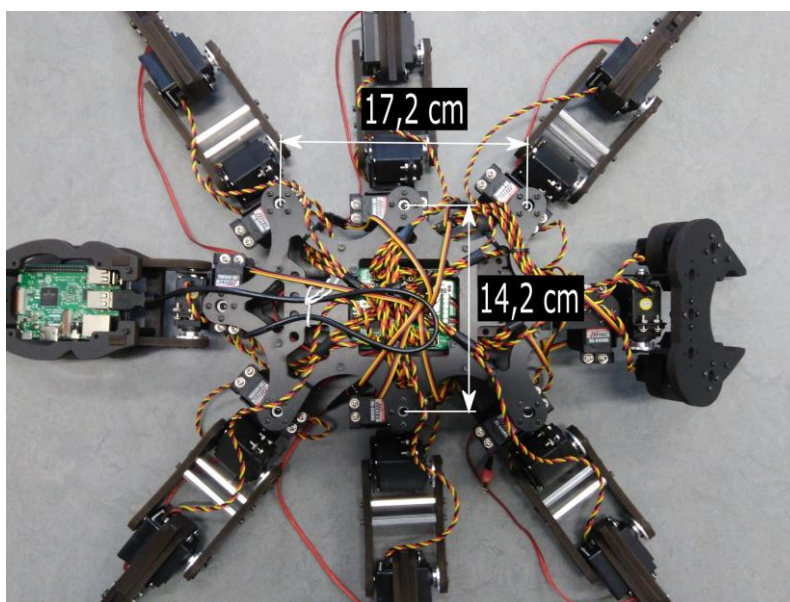
A-Pod má na každé noze tři serva a tím i stupně volnosti. Servo nejbliže u těla zajišťuje pohyb nohy horizontálně (dopředu a dozadu), zatímco další dvě serva se pohybují vertikálně (nahoru a dolů).

Na hlavě je celkem pět serv. Dvě ovládají zavírání a otevírání klepet. Zbývá zajišťují horizontální, vertikální a otočný pohyb celou hlavou.

V ocasu jsou dvě serva. Jedno se pohybuje vertikálně, druhé horizontálně.

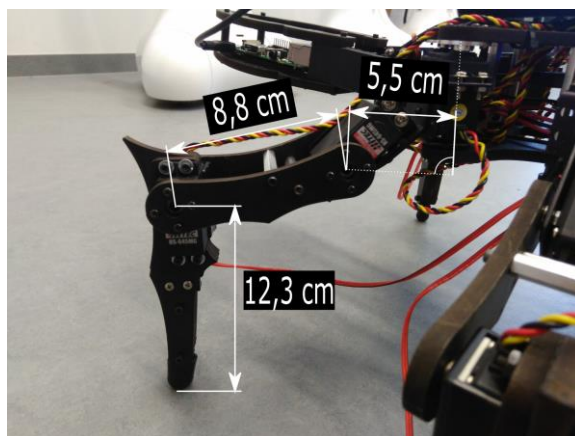
#### 3.1.3 Rozměry

Rozměry těla potřebné pro výpočty v této práci jsou uvedeny na obrázku 3.2. Střed těla je umístěn v polovině vzdálenosti mezi středním párem nohou.



Obrázek 3.2: Rozměry těla A-Podu

Dalšími důležitými rozměry jsou délky článků nohy, které je možné vidět na obrázku 3.3.



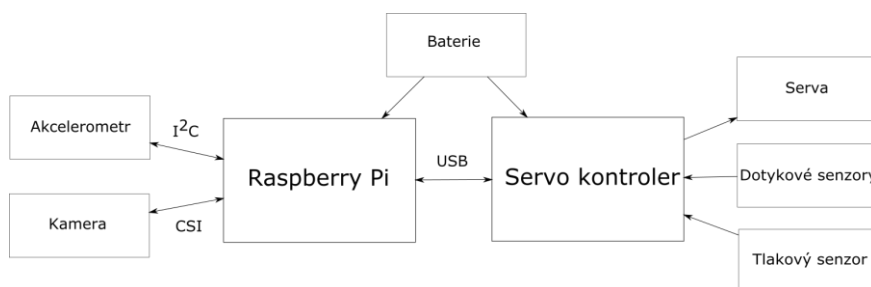
Obrázek 3.3: Rozměry nohy A-Podu

## 3.2 Elektronika

### 3.2.1 Propojení součástek

Schéma zapojení součástek je na obrázku 3.4 a její umístění na obrázku 3.5

Hlavní řídicí jednotkou hexapodu je počítač Raspberry Pi umístěný v ocasu. Ten přes USB rozhraní posílá příkazy servo kontroleru SSC-32u od firmy Lynxmotion, který se stará o nastavování a udržování pozice všech serv. Obě desky jsou napájeny z baterie. Dotykové senzory na nohách a tlakový senzor v klepetech jsou zapojeny do servo kontroleru. K Raspberry Pi je připojen akcelerometr přes rozhraní I<sup>2</sup>C a kamera přes rozhraní CSI.

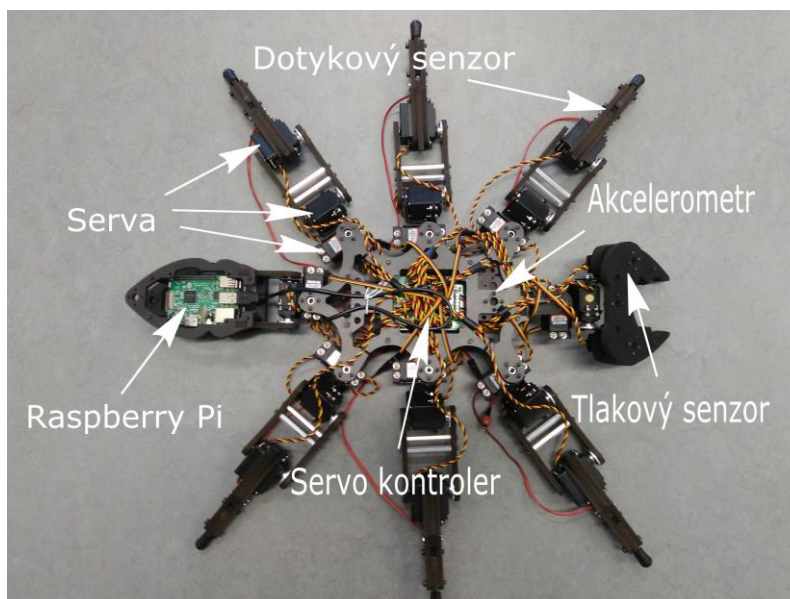


Obrázek 3.4: Schéma zapojení elektroniky

### 3.2.2 Raspberry Pi

Hlavní řídicí jednotkou robota je Raspberry Pi (RPi) model 3B. Jedná se o malý počítač velikosti kreditní karty. Byl vyvinut společností Raspberry Pi Foundation pro výukové účely. [12] Cílem bylo zpřístupnit programování širší

### 3. KONKRÉTNÍ HEXAPOD



Obrázek 3.5: Umístění elektroniky

skupině lidí. K tomu napomáhá nízká cena počítače a jeho univerzálnost. Jeho technické parametry jsou vypsány v tabulce 3.1.

Tabulka 3.1: Technické parametry Raspberry Pi 3B [12]

Procesor	Quad Core 1.2GHz Broadcom BCM2837 64bi
Paměť	1GB
Rozhraní	4x USB 2
	WiFi
	40 GPIO pinů - včetně I <sup>2</sup> C
	CSI port pro kameru
	Ethernet
	HDMI

Na SD kartě je nainstalován operační systém Raspbian.

Při práci se k RPi připojujeme po WiFi vzdáleně přes protokol SSH. Bezdrátové připojení je důležité, aby se nenarušila mobilita robota.

Při řešení problémů je možné připojit k RPi monitor, klávesnici a myš a případně ethernetový kabel.

Do jednoho z USB portů je připojený servo kontroler, který se připojuje jako /dev/ttyUSB0 zařízení.

Kamera je zapojena do CSI portu, který se nachází vedle HDMI výstupu. Návod na zapojení lze nalézt na stránce [13].

Akcelerometr je připojen přes I<sup>2</sup>C GPIO piny.

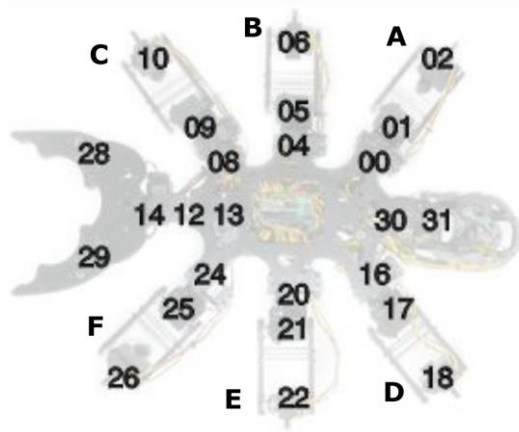


Deska má dostatečný výpočetní výkon a paměť pro výpočty navržené v rámci práce.

### 3.2.3 Servo kontroler

V robotovi je použit servo kontroler SSC-32U USB Servo Controller Board od firmy Lynxmotion, jehož dokumentaci lze najít na stránce [14]. Základním úkolem kontroleru je ovládání serv pomocí PWM signálu (pulzně šířkové modulace). Dále jsou do něj zapojeny dotykové a tlakové senzory.

Deska dokáže ovládat až 32 serv současně a přijímat analogový signál z 8 zdrojů. Serva se zapojují do pinů 0-31 a číslo pinu odpovídá id serva, které je posíláno v rámci příkazů (obr. 3.6). Senzory se zapojují do pinů A-H.



Obrázek 3.6: Indexy pinů, do kterých jsou zapojená serva (čísla) a dotykové senzory (písmena) [15]

Kontroler není sám o sobě programovatelný, ale přijímá příkazy od RPi po USB rozhraní. Součástí firmwaru je i předpřipravený program umožňující chůzi hexapodu s 12 stupni volnosti (2 na nohu), takže pro A-Pod se 3 stupni volnosti není použitelný.

Nejdůležitějším příkazem kontroleru je příkaz pro změnu polohy serva. Jeho struktura je popsána na obrázku 3.7. Pro lepší přehlednost mohou být v příkazu použity mezery, tabulátory a odřádkování, protože jsou kontrolerem v tomto příkazu ignorovány.

Příkazy pro nastavování serv mohou být sdružovány. Díky tomu lze jedním příkazem nastavovat celou skupinu serv.

Pohyb lze časovat dvěma způsoby. Jednak je možné specifikovat, za jaký časový úsek má být pohyb proveden. Tato informace je sdílena mezi všechna serva při skupinovém nastavování. To znamená, že všechna serva začnou a skončí pohyb současně, přesně po určené době. Druhým způsobem je určení rychlosti změny šířky pulzu. Tento způsob ovlivňuje pouze servo bezprostředně

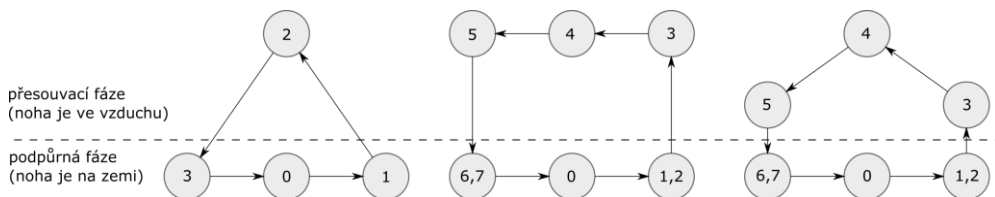
## 4.1 Řízení chůze

Knihovna zařizuje pohyb robota z místa na místo. Při volání funkce pro pohyb uživatel zadává parametry vzdálenosti a úhlu, podle kterých knihovna vybírá typ pohybu. Následně navrhuje dráhu, po které robot půjde, a vypočítává, jak se musí pohybovat nohy, aby se robot pohyboval po zvolené dráze. Vypočítané pozice nohou jsou knihovnou přiřazeny k jednotlivým fázím cyklu nohou, kterými se postupně prochází. V každé fázi cyklu jsou posílány příslušné příkazy pro serva, dokud robot nedorazí na místo.

### 4.1.1 Dráha nohou a jejich synchronizace

Zvolila jsem typ chůze tripod (viz kapitola 2.2.3), protože je díky své univerzálnosti základní dovedností hexapodů. Jeho nedostatek v podobě menší stability je pro základní chůzi v nepříliš členitém terénu zanedbatelný.

Každá noha se pohybuje v cyklu rozděleném do několika částí (stavů) (obr. 4.1). Stavů ve spodní linii obrázku jsou součástí podpůrné fáze cyklu nohy (noha je na zemi a její pohyby hýbou tělem), zbylé patří do přesouvací fáze (noha je ve vzduchu a přesouvá se na příhodné místo, pro začátek dalšího kroku). Základem každého cyklu je stav 0, kdy se noha nachází ve své výchozí pozici, jejíž  $x$  a  $y$  souřadnice jsou pro ni během programu neměnné. Stav 0 je středem podpůrné fáze dané nohy.



Obrázek 4.1: Cyklus nohy, základní (vlevo), terénní (uprostřed), vylepšený (vpravo)

Na obrázku 4.1 vlevo je zobrazen základní cyklus nohy pro chůzi po rovině. Základem je stav 0. Stavů 1 a 3 umožňují robotovi pohyb. Stav 2 umožňuje přesun vzduchem. Bohužel kvůli pohyblivému konci nohy, který se stlačuje zhruba o 3 mm, se noha začala mezi stavy 2 a 3 třít o zem a zastavila se dřív, než došla na určené místo, což způsobovalo nepřesnost chůze. Proto byl cyklus nahrazen.

Cyklus nohy při chůzi v terénu je zobrazen na obrázku 4.1 uprostřed. Oproti základní chůzi přibýly další dvě části cyklu, které zajišťují kolmost pohybu směrem dolů, takže ať je noha zastavena kdykoli během této části, nezmění se délka kroku.

Vylepšený cyklus pro chůzi po rovině, který je možné vidět na obrázku 4.1 vpravo, vznikl upravením terénního cyklu a nahradil základní variantu. Stavby 3 a 5 jsou v něm sníženy, protože stačí, aby se noha pohybovala kolmo k zemi jen těsně nad ní a ne při celém pohybu směrem dolů.

Kvůli typu chůze tripod, kdy mají nohy v různých skupinách fázový posun v cyklu o  $\frac{1}{2}$ , jsou do cyklů přidány nadbytečné stavy, aby se mohly inkrementovat fáze všech nohou najednou. Tedy například u terénního cyklu jsou skupiny nohou v jednu chvíli ve fázích 0 a 4, 1 a 5, 2 a 6, ..., 7 a 3. Všechny nohy se z pozice do pozice přesouvají pomocí skupinového příkazu pro serva (natočení serv je vypočítáno pomocí inverzní kinematiky v kapitole 4.3) za jednotný čas.

### 4.1.2 Trajektorie

Při pohybu robota z místa na místo je důležité určit dráhu, po které robot půjde. Následně je potřeba určit pohyby nohou v podpůrné fázi, aby se robot pohyboval po zvolené dráze.

Hexapod bude umět tři základní druhy pohybu:

- chůze po přímce ve všech směrech,
- otočení se na místě,
- chůze po křivce.

V každém z případů se tělo pohybuje jiným způsobem, což je nutné zohlednit při tvorbě trajektorie nohou, a je tedy třeba přistupovat k nim samostatně. V následující části práce rozeberu podrobněji všechny tři případy.

#### 4.2.2.1 Chůze po přímce

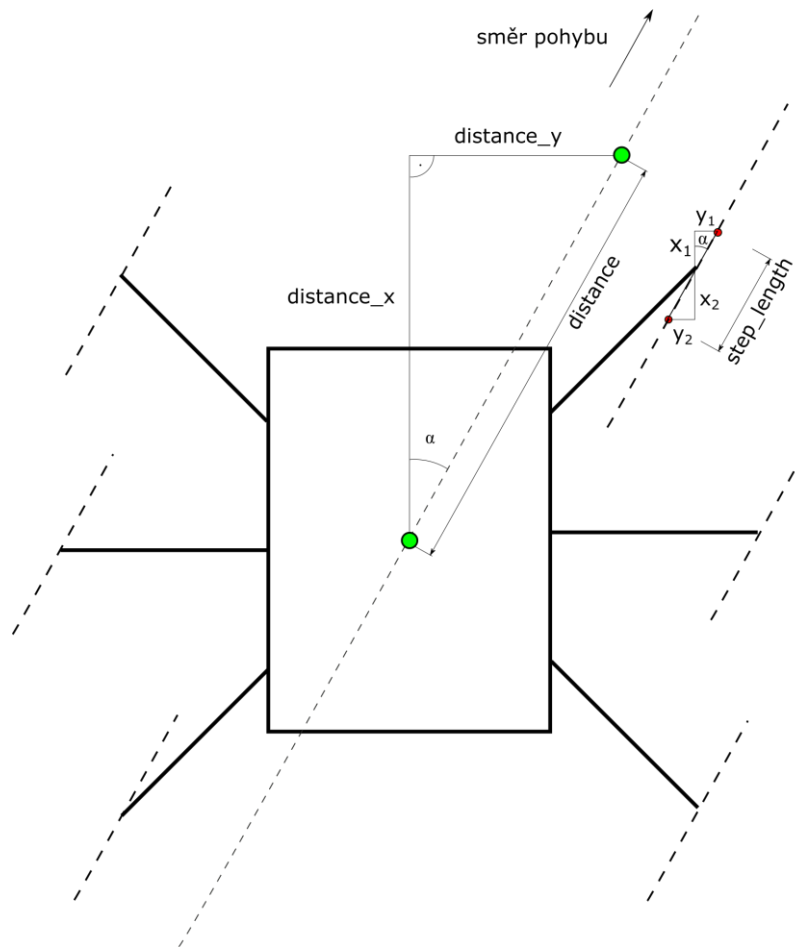
Při chůzi po přímce se robot pohybuje přímo k cíli, aniž by se v průběhu otáčel. Aby se robot pohyboval daným způsobem, musí se jeho nohy v podpůrné fázi také pohybovat po přímkách rovnoběžných s trajektorií těla. Zároveň musí být délka kroku stejná pro všechny nohy, které jsou současně v podpůrné fázi.

Přímka, po které robot jde, je určena přeponou trojúhelníku (distance), jehož odvěsny jsou tvořeny vzdáleností ve směru x (distance\_x) a y (distance\_y) (obrázek 4.2). Přeponou je určen směr i vzdálenost, kterou má robot ujít.

Při určování délky kroku (step\_length) jsem zvolila variantu, že všechny kroky jsou v rámci jednoho zavolání funkce stejné. Druhou variantou by byla fixní délka kroku při každém volání a jen poslední krok by byl upraven, aby součet délek kroků odpovídal požadované délce přesunu. Při výpočtu délky kroku je nejdříve určen počet kroků pomocí vydělení délky dráhy robota délkou ideálního kroku, která byla stanovena experimentálně. Délka jednoho kroku je vypočtena vydělením délky dráhy robota počtem kroků.

#### 4. NÁVRH

Polohy nohou v jednotlivých fázích kroku jsou určeny z délky kroku a úhlu ( $\alpha$ ), který svírá přepona směr chůze s osou x.



Obrázek 4.2: Schéma chůze po přímce

##### 4.2.2.2 Otočení na místě

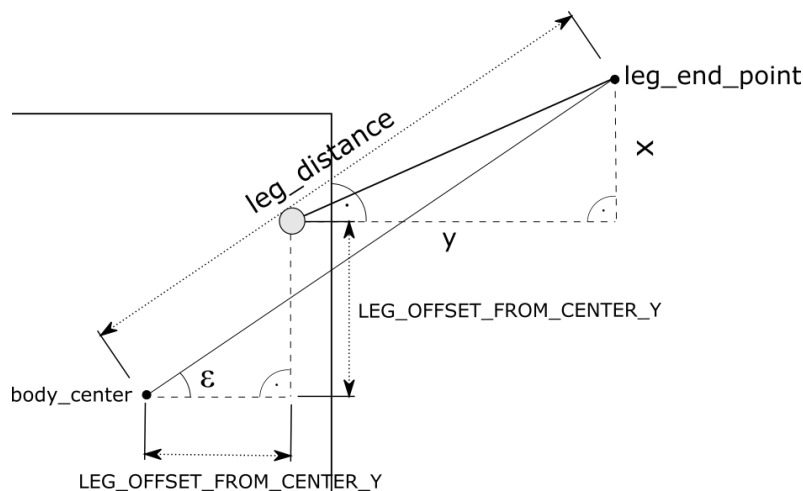
Pro docílení otočení robota na místě je nutné, aby se nohy pohybovaly po kružnici se středem ve středu těla. Nohy, které jsou současně v podpůrné fázi, se musí pohybovat se stejnou úhlovou rychlostí. Při zadání kladného úhlu se robot otáčí proti směru hodinových ručiček.

Základem je vypočítat vzdálenost ( $leg\_distance$ ) koncových bodů nohou ( $leg\_end\_point$ ) od středu těla ( $body\_center$ ) v základní pozici a úhel ( $\epsilon$ ) tohoto spojení od kolmice na tělo. K výpočtu je třeba vzdálenost serva o od středu těla ve složkách x ( $LEG\_OFFSET\_FROM\_CENTER\_X$ ) a y ( $LEG\_OFFSET\_FROM\_CENTER\_Y$ ). K ní je po složkách přičtena sou-

řadnice koncového bodu nohy (vzdálenost konce nohy a serva o – x, y) a následně použita Pythagorova věta (obrázek 4.3).

Velikost úhlu, o který se robot otočí za jeden krok, je vypočtena obdobně jako délka kroku při chůzi po přímce.

Pro nalezení trajektorií nohou je aplikováno následující: noha se musí pohybovat po kružnici kolem středu těla, takže vzdálenost koncového bodu a středu těla je konstantní. Nohy mají mít stejnou úhlovou rychlost, proto se během jednoho kroku musí otočit o stejný úhel. Pozice nohou jsou vypočteny tak, že k úhlu základní pozice je přičtena/odečtena polovina úhlu kroku ( $\sigma$ ). Jsou vypočítány složky x a y vzdálenosti koncového bodu v krajních pozicích (`leg_end_point_new`) a odečtena po složkách vzdálenost základního postavení. Tím je určena vzdálenost krajních poloh od základního postavení nohy ( $x_2, y_2$ ) (obrázek 4.4).



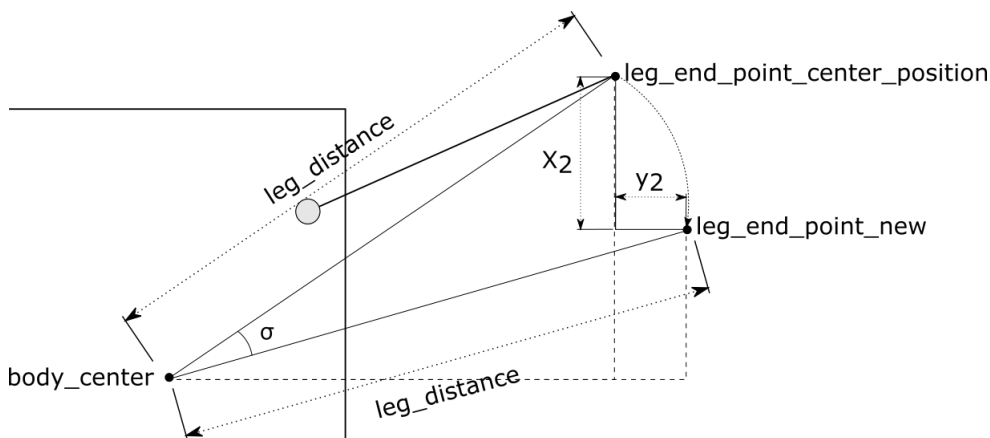
Obrázek 4.3: Schéma pro výpočet vzdálenosti a úhlu konce nohy od středu těla

#### 4.2.2.3 Chůze po křivce

Chůzi po křivce lze využít při přesouvání robota mezi dvěma body, kdy ve výsledném bodě má být natočen jinak než v původním. Tento případ lze vyřešit i přesunem robota po přímce a následným otočením na místě, ale pohyb po křivce je ve většině případů přirozenější.

Vhodnou křivkou, po které se může robot pohybovat, je Bézierova křivka. Přestože patří mezi aproximační křivky, je u ní zaručeno, že prochází počátečním a koncovým řídicím bodem. Zároveň díky řídicím bodům sousedícím s koncovými body lze specifikovat natočení v koncových bodech. Vzhledem k těmto vlastnostem je možné navazovat více křivek na sebe. Pro základní

#### 4. NÁVRH



Obrázek 4.4: Schéma krajní polohy nohy při otáčení na místě

užití stačí kubická Bézierova křivka, která ale při určitých parametrech nezajišťuje přirozený pohyb hexapodu.

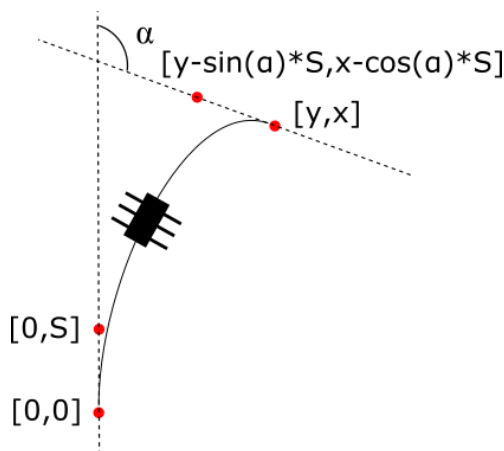
Koncové body křivky jsou určeny aktuální polohou robota, která je brána jako bod  $[0, 0]$ , a zadanou vzdáleností určující bod  $[y, x]$ . Natočení robota v koncovém bodě je dáno zadaným úhlem. Ve směru natočení robota jsou ve vzdálenosti  $S$  (experimentálně zvolená hodnota, aby se robot ke směru přimyká dostatečně dlouho) od koncových bodů vytvořeny zbývající dva řídicí body. Tím jsou zadány všechny čtyři body kubiky a lze jí pomocí změny parametru  $t$  (čas) procházet (obr. 4.5). V  $t = 0$  se robot nachází v počátečním bodě, v  $t = 1$  se robot nachází v koncovém bodě. V rovnici 4.1 je vidět dosazení řídicích bodů do maticového vyjádření Bézierovy křivky.

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 \\ 0 & S \\ y - \sin(\alpha) * S & x - \cos(\alpha) * S \\ y & x \end{bmatrix} \quad (4.1)$$

Zderivováním rovnic podle  $t$  a jeho následným dosazením je získáno natočení robota v podobě tečného vektoru v čase  $t$ .

Rovnice 4.2 a 4.3 převzaté z [18] ukazují použití transformační matice, která bod  $[x, y]$  přesune ve směru  $[h, k]$  a následně otočí o úhel  $\alpha$ .

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & h * \cos(\alpha) - k * \sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) & h * \sin(\alpha) + k * \cos(\alpha) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.2)$$



Obrázek 4.5: Návrh Bézierovy křivky

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & -h \\ -\sin(\alpha) & \cos(\alpha) & -k \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4.3)$$

Pro tvorbu trajektorie nohou navrhuji přes křivku procházet po půlkrocích robota. Parametr  $t$  je volen tak, aby rozdíl mezi současnou a následující pozicí byl přiměřený jednomu půlkroku (jeho délkou a otočením robota).

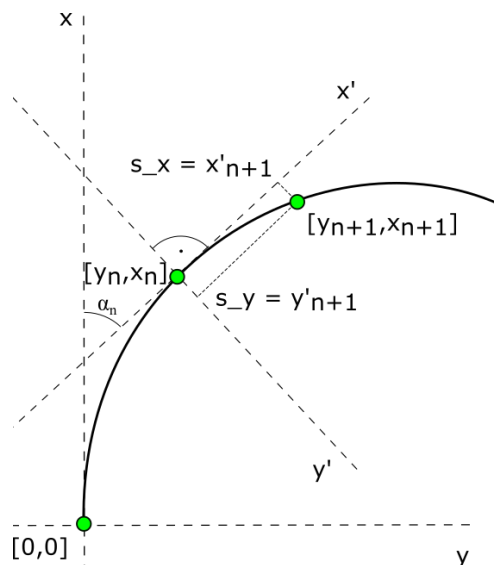
Pro výpočet polohy nohou v každém půlkroku je třeba nejdřív vypočítat délku půlkroku těla po složkách ( $s_y$ ,  $s_x$ ) a úhel ( $s_\alpha$ ), o který má být robot otočen. Obě informace jsou vyjádřeny relativně k poloze, ve které se nachází po posledním půlkroku (obr. 4.6). Výpočet je uveden v rovnicích 4.4 a 4.5

$$\begin{pmatrix} s_y \\ s_x \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(-\alpha_n) & -\sin(-\alpha_n) & -y_n * \cos(-\alpha_n) + x_n * \sin(-\alpha_n) \\ \sin(-\alpha_n) & \cos(-\alpha_n) & -y_n * \cos(-\alpha_n) - x_n * \sin(-\alpha_n) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{n+1} \\ x_{n+1} \\ 1 \end{pmatrix} \quad (4.4)$$

$$s_\alpha = \alpha_{n+1} - \alpha_n \quad (4.5)$$

Výpočet posunutí nohou při půlkroku se dělí na 2 varianty:

1. Noha je ve středové poloze  $[y_c, x_c]$  podpůrné fáze a přesouvá se do zadní polohy  $[y_2, x_2]$ . Tedy souřadnice středové polohy je potřeba transformovat způsobem uvedeným v rovnici 4.6.
2. Noha je v přední poloze  $[y_1, x_1]$  a přesouvá se do středové polohy  $[y_c, x_c]$  podpůrné fáze. Tedy souřadnice středové polohy jsou transformované a původní souřadnice jsou získány pomocí výpočtu v rovnici 4.7.



Obrázek 4.6: Chůze po křivce: Vyjádření souřadnic půlkroku relativně k poloze těla po posledním půlkroku

V případě výpočtu souřadnic levých nohou se před výpočtem musí u y souřadnice změnit znaménko, aby byly tyto souřadnice na správné straně osy (pro všechny nohy jsou za normálních okolností y souřadnice kladné). Po výpočtu se zase převedou zpět.

$$\begin{pmatrix} y_2 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(-s_\alpha) & -\sin(-s_\alpha) & -s_y * \cos(-s_\alpha) + s_x * \sin(-s_\alpha) \\ \sin(-s_\alpha) & \cos(-s_\alpha) & -s_y * \cos(-s_\alpha) - s_x * \sin(-s_\alpha) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_c \\ x_c \\ 1 \end{pmatrix} \quad (4.6)$$

$$\begin{pmatrix} y_1 \\ x_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(-s_\alpha) & \sin(-s_\alpha) & s_y \\ -\sin(-s_\alpha) & \cos(-s_\alpha) & s_x \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_c \\ x_c \\ 1 \end{pmatrix} \quad (4.7)$$

V obou případech se od výsledných souřadnic odečte původní souřadnice středové polohy, aby se získal jejich offset.

#### 4.1.3 Chůze v terénu

Při chůzi v terénu je třeba přizpůsobit cyklus nohy prostředí. K tomu slouží zpětná vazba z dotykových senzorů na nohách, které se sepnou, když je noha na zemi. Informace z nich používám při pokládání nohy na zem, kdy kontroluji stav senzoru. Když se pne, nohu zastavím. Pro zastavení serva existují dvě možnosti. Buď použít funkci Stop, kterou servo kontroler má. Nebo je možné zjistit od servo kontroleru aktuální polohu serva a na ně serva nastavit. Více



se mi osvědčila druhá varianta, protože při použití funkce Stop sebou serva trhaly nazpátek, že se noha ani nedotýkala podložky.

Ve chvíli, kdy se zastaví všechny nohy v pohybu dolů, se vypočtou nové offsety výšky těla pro tyto nohy. Výšky se využijí ve zbytku aktuálního průběhu cyklu nohou, aby bylo stále počítáno s tím, že noha má být zvednuta více než obvykle, aby správně přiléhala k zemi.

Následně proběhne upravení výšek všech nohou. Úprava se skládá ze dvou částí:

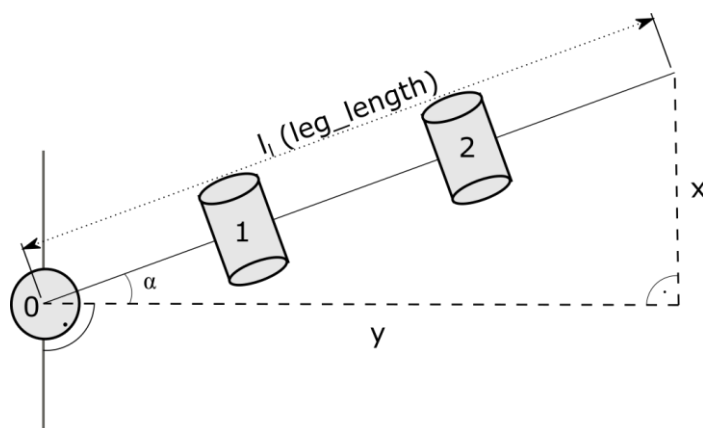
- Srovnání těla do vodorovné polohy. K tomu je využit akcelerometr, díky kterému se zjistí natočení těla v x a y složce. Následně se vypočte potřebná regulace výšky nohou, aby bylo tělo vodorovné.
- Vyrovnání těla do standardní výšky. Tato úprava je udělána přičtením rozdílu standardní výšky a aritmetického průměru výšek nohou k výškám nohou.

## 4.2 Odvození inverzní kinematiky

V této části práce popíši, jak jsem odvodila vzorce pro inverzní kinematiku nohy. Cílem je pro zadané souřadnice koncového bodu nohy najít odpovídající natočení serv.

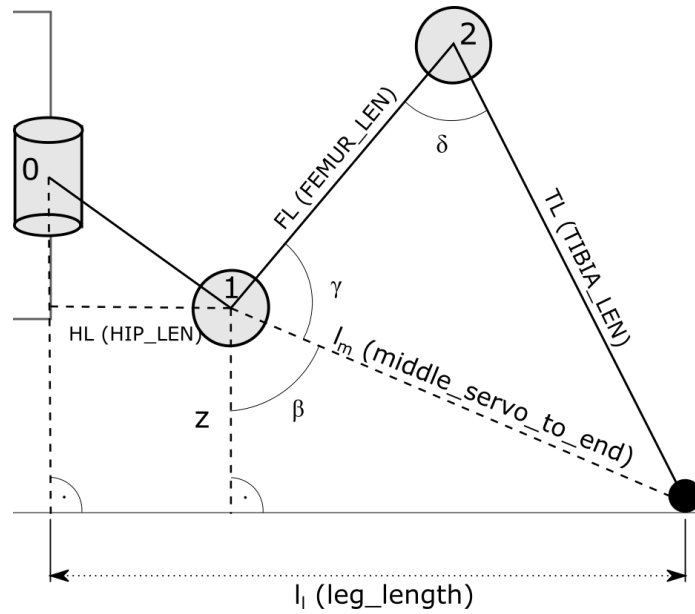
Poté, co jsem návrh vytvořila sama a měla hotovou i implementaci, jsem při pokračování v práci narazila na několik stránek, které problematiku řeší stejně. Například: [19], [20], [21].

Obrázky 4.7 a 4.8 ukazují půdorys a bokorys daného problému pro jeho lepší pochopení. Šedé kruhy a válce na nich představují serva. Plné černé čáry jsou části končetiny. Konstanty jsou uvedeny velkými písmeny, zatímco proměnné malými.



Obrázek 4.7: Inverzní kinematika - půdorys

#### 4. NÁVRH



Obrázek 4.8: Inverzní kinematika - bokorys

Každá noha je ve svém vlastním prostoru a její inverzní kinematika je počítána nezávisle na poloze těla a ostatních nohách. Vzorce jsou pro všechny nohy stejné. Přestože jsou krajní páry nohou ve střední poloze serv vytočeny o 45° od středového páru, při výpočtech je uvažována základní poloha nohy kolmo k tělu.

Na začátku výpočtu jsou známy pouze konstanty v podobě délek článků nohy (HIP\_LEN, FEMUR\_LEN, TIBIA\_LEN) a požadovaných souřadnic koncového bodu ( $x$ ,  $y$ ,  $z$ ). HIP\_LEN je vzdálenost nultého a prvního serva se zanedbáním výškového rozdílu, protože nulté servo umožňuje pohyb vodorovně, takže výškový rozdíl těchto dvou serv je neměnný.

Ze souřadnic  $x$  a  $y$  je vypočten úhel  $\alpha$  (rovnice 4.8) a hodnota proměnné leg\_length (rovnice 4.9) (obr. 4.7).

$$\alpha = \arctan \frac{x}{y} \quad (4.8)$$

$$l_l = \sqrt{x^2 + y^2} \quad (4.9)$$

Dále je určen úhel  $\beta$  (rovnice 4.10) a proměnná middle\_servo\_to\_end (rovnice 4.11) (obr. 4.8)

$$\beta = \arctan \frac{(l_l - HL)}{\sqrt{\quad}} \quad (4.10)$$

$$l_m = \sqrt{\quad + (l_l - HL)^2} \quad (4.11)$$

---

### 4.3. Odvození inverzní kinematiky

Díky vypočtení `middle_servo_to_end` je známa délka všech stran v horním trojúhelníku na obrázku 4.8, a tedy lze pomocí Kosinové věty vypočítat úhel  $\gamma$  (rovnice 4.12) a  $\delta$  (rovnice 4.13)

$$\gamma = \frac{l_m^2 + FL^2 - TL^2}{2 * l_m * FL} \quad (4.12)$$

$$\delta = \frac{FL^2 + TL^2 - l^2}{2 * FL * TL} \quad (4.13)$$

Nakonec jsou převedeny úhly na offsety úhlů od středních poloh jednotlivých serv.

$$\text{servo\_0\_angle\_offset} = \alpha \quad (4.14)$$

$$\text{servo\_1\_angle\_offset} = \beta + \gamma - 90 \quad (4.15)$$

$$\text{servo\_2\_angle\_offset} = \delta - 90 \quad (4.16)$$

## 6.1 Chůze po rovině

Při testování chůze po rovině jsem samostatně testovala chůzi po přímce, otočení na místě i chůzi po křivce. Pro každou možnost jsem udělala sérii testů, při kterých jsem pozorovala a měřila, jak robot chodí.

Chůzi po přímce jsem testovala ve dvou úrovních. Jednak jsem vyzkoušela, že robot chodí, a to ve směru zadaném uživatelem. Robot se při všech pokusech pohyboval plynule definovaným směrem.

Druhý test ověřoval přesnost chůze vpřed a vzad. Měřením jsem zjistila, že neujde přesně vzdálenost, která mu byla zadána. Cestou dopředu robot vzdálenost přecházel, dozadu ji naopak neušel. Odchylka se různila podle zvolené délky kroku, ale pro jedno nastavení byla stále stejná. V návaznosti na tyto výsledky jsem hledala chybu v programu, ke které tato systematickост navádí. Avšak nic jsem nenašla. Posílané příkazy pro serva jsou shodné pro chůzi dozadu i dopředu, jen se posílají v opačném pořadí. Jako možné příčiny odhaduji nepřesné smontování robota a nepřesnost serv a jejich nedostatečně dobrou kalibraci. Tu je obtížné provést a je třeba ji pravidelně opakovat. Dále se materiál, ze kterého je robot vyroben, výrazně ohýbá při zátěži. Tyto problémy se vyskytují i u podstatně kvalitnějších robotů.

Při otáčení na místě se robot skutečně otočí a jeho střed těla zůstane na stejném místě. Stejně jako u chůze po přímce má robot problém s přetáčením, které je v obou směrech pohybu stejné.

## 5. TESTOVÁNÍ

---

Při testování chůze po křivce jsem zkoušela několik různých tvarů a jejich směrů. Ve všech případech se robot na pohled pohyboval správně po zadané křivce. Ve chvíli, kdy robot procházel ostřejší zatačkou, jeho pohyb nebyl podle očekávání úplně plynulý a přirozený. Stejně jako u dalších druhů pohybu se robot přesunul do větší vzdálenosti, než měl zadáno, avšak se zachováním tvaru křivky.

### 6.2 Chůze v terénu

Chůze v terénu byla testována na dvou typech překážek. Jednou byly krabice o výšce 5 cm a druhou nakloněná rovina s úhlem  $12^\circ$ .

Krabice, které byly uloženy pouze v jedné vrstvě, dokázal robot překonat. Jediné problémy nastávaly, když došlápl na okraj krabice. Noha se při dotyku s krabicí zastavila, ale když na ni byla přenesena váha, tak se sklouzla přes hranu. To způsobilo rozhození robota, než dokončil krok, ale poté bylo tělo opět vyrovnáno díky použití akcelerometru.



Obrázek 6.1: Chůze robota po nakloněné rovině